

**APPLICATION
FOR
UNITED STATES LETTERS PATENT**

APPLICANT NAME: Holden et al.

TITLE: METHOD, SYSTEM AND PROGRAM PRODUCT FOR
REPLICATING MESSAGE STATUS CHANGES ACROSS
MESSAGING SYSTEMS

DOCKET NO.: LOT920030051US1

INTERNATIONAL BUSINESS MACHINES CORPORATION

CERTIFICATE OF MAILING UNDER 37 CFR 1.10

I hereby certify that, on the date shown below, this correspondence is being deposited with the United States Postal Service in an envelope addressed to the Mail Stop Patent Application, Commissioner for Patents, Alexandria, VA 22313-1450 as "Express Mail Post Office to Addressee" Mailing Label No. EV225574527US

on December 8, 2003

Dorothea Rubbone

Name of person mailing paper

Dorothea Rubbone

Signature

December 8, 2003

Date

METHOD, SYSTEM AND PROGRAM PRODUCT FOR REPLICATING MESSAGE STATUS CHANGES ACROSS MESSAGING SYSTEMS

Background of the Invention

1. Field of the Invention

[0001] In general, the present invention provides a method, system and program product for replicating message status changes across (replica) messaging systems. In addition, the present invention provides a method, system and program product for maintaining log chronology for message status changes replicated across the messaging systems.

2. Related Art

[0002] As the use of electronic messaging such as electronic mail becomes more pervasive, there is a growing need to provide replication among electronic mail systems. Specifically, due to the distributed nature of many of today's computing infrastructures, many organizations could benefit from the redundancy provided in having multiple replicas of its electronic mail system. In such a system, each such replica could contain a database that tracks the activity of each user. For example, as a user of system "A" reads a message, a corresponding entry could be made in the database of system "A." A similar entry would be made if the user later marked the same message as "unread." If system "B" is a replica of system "A," such entries should be replicated thereto. Unfortunately, no existing system currently provides an optimal way to make such replication. That is, no existing system provides a way to optimally ensure that each replica has an identical

view of each user's activity. This is especially the case, for example, where a user changes a status of a message to "read" on system "A" and later to "unread" on system "B." In such a scenario, there currently no way to ensure that the both systems reflect identical versions of the most recent information.

[0003] Additional problems would exist in maintaining log chronology for message status changes. Specifically, as is known, the clocks of computer systems within a distributed network are often not synchronized. Accordingly, the clock for system "A" could report a time of 1:00 PM, while the clock for system "B" could report a time of 1:10 PM. With such differences, there is a chance that a more recent status change on a particular system could be improperly "trumped" by a previous status change replicated from another system. This is yet another issue no current system addresses.

[0004] In view of the foregoing, there exists a need for a method, system and program product for replicating message status changes across (replica) messaging systems. In addition, the need exists for a method, system and program product for maintaining log chronology for message status changes replicated across the messaging systems.

Summary of the Invention

[0005] In general, the present invention first provides a method, system and program product for replicating message status changes across (replica) messaging systems.

Specifically, under the present invention, when a status of a message for a user is changed on a first messaging system, a corresponding entry is made in a log associated with the user on the first messaging system. Thereafter, the entry is communicated to a second

messaging system that also has a log associated with the user. Before the entry is formally replicated, however, it is first determined whether the status change is more recent than any other status changes that might exist for the same message with the log of the second messaging system. If so, the entry is replicated. If not, the entry is discarded from the second messaging system. In addition, the present invention provides a method, system and program product for maintaining log chronology for message status changes replicated across the systems. Specifically, when status change is entered into the log of the first messaging system, it is entered along with a clock time of the first messaging system. Thereafter, the entry containing the status change and the clock time is replicated to the second messaging system. If the entry is to be logged in the log of the second messaging system, it is logged along with the clock time of the first messaging system as well as a clock time of the second messaging system (i.e., that corresponds to the clock time of the first messaging system when the status change occurred).

[0006] A first aspect of the present invention provides a method for replicating message status changes across messaging systems, comprising: changing a status of a message for a user on a first messaging system; entering the status change into a log associated with the user on the first messaging system; communicating the status change to a second messaging system, wherein the second messaging system is a replica of the first messaging system; and determining, on the second messaging system, if the status change is more recent than any other status changes of the message within a log associated with the user on the second messaging system.

[0007] A second aspect of the present invention provides a method for replicating message status changes across messaging systems, comprising: providing a first messaging system having a first set of logs corresponding to a set of users, wherein the first set of logs includes entries reflecting status changes for electronic messages received by the set of users; communicating the first set of logs to a second messaging system having a second set of logs corresponding to the set of users; and determining, on the second messaging system, if the entries within the first set of logs are more recent than existing entries within the second set of logs.

[0008] A third aspect of the present invention provides a system for replicating message status changes across messaging systems, comprising: a log entry system for entering a status change of a message for a user into a log associated with the user on a first messaging system; and a replication system for communicating the status change to a second messaging system, wherein the second messaging system includes a precedence system for determining whether the status change for the message is more recent than any other status changes for the message previously entered into a log associated with the user on the second messaging system.

[0009] A fourth aspect of the present invention provides a system for replicating message status changes across messaging systems, comprising: a log entry system for entering status changes of messages for a user into a log associated with the user on a first messaging system; a replication system for communicating the status changes a second messaging system; a precedence system for determining whether a status change for a particular message received from the second messaging system is more recent than any

status changes for the particular message previously entered into the log; and a table maintenance system for maintaining an unread table on the first messaging system that identifies any messages for the user that are unread.

[0010] A fifth aspect of the present invention provides a program product stored on a recordable medium for replicating message status changes across messaging systems, which when executed comprises: program code for entering a status change of a message for a user into a log associated with the user on a first messaging system; and program code for communicating the status change to a second messaging system, wherein the second messaging system includes program code for determining whether the status change for the message is more recent than any other status changes for the message previously entered into a log associated with the user on the second messaging system.

[0011] A sixth aspect of the present invention provides a method for maintaining log chronology for message status changes replicated across messaging systems, comprising: changing a status of a message for a user on a first messaging system; entering the status change into a log associated with the user on the first messaging system, wherein the status change is entered into the log along with a clock time of the first messaging system; communicating the status change and the clock time of the first messaging system to a second messaging system; and entering the status change into a log associated with the user on the second messaging system, wherein the status change is entered into the log associated with the user on the second messaging system with the clock time of the first messaging system and a clock time of the second messaging system.

[0012] A seventh aspect of the present invention provides a system for maintaining log chronology for message status changes replicated across messaging systems, comprising: a log entry system for entering a status change of a message for a user into a log associated with the user on a first messaging system, wherein the status change is entered into the log along with a clock time of the first messaging system; and a replication system for communicating the status change and the clock time of the first messaging system to a second messaging system, wherein the status change is entered into a log associated with the user on the second messaging system along with the clock time of the first messaging system and a clock time of the second messaging system.

[0013] An eighth aspect of the present invention provides a program product stored on a recordable medium for maintaining log chronology for message status changes replicated across messaging systems, which when executed comprises: program code for entering a status change of a message for a user into a log associated with the user on a first messaging system, wherein the status change is entered into the log along with a clock time of the first messaging system; and program code for communicating the status change and the clock time of the first messaging system to a second messaging system, wherein the status change is entered into a log associated with the user on the second messaging system along with the clock time of the first messaging system and a clock time of the second messaging system.

[0014] Therefore, the present invention first provides a method, system and program product for replicating message status changes across (replica) systems. The present

invention also provides a method, system and program product for maintaining log chronology for message status changes replicated across the systems.

Brief Description of the Drawings

[0015] These and other features of this invention will be more readily understood from the following detailed description of the various aspects of the invention taken in conjunction with the accompanying drawings in which:

[0016] Fig. 1 depicts an illustrative replication architecture according to the present invention.

[0017] Fig. 2 depicts a more detailed diagram of the illustrative replication architecture of Fig. 1.

[0018] Fig. 3 depicts a first method flow diagram according to the present invention.

[0019] Fig. 4 depicts a second method flow diagram according to the present invention.

[0020] It is noted that the drawings of the invention are not necessarily to scale. The drawings are merely schematic representations, not intended to portray specific parameters of the invention. The drawings are intended to depict only typical embodiments of the invention, and therefore should not be considered as limiting the scope of the invention. In the drawings, like numbering represents like elements.

Detailed Description of the Invention

[0021] As indicated above, the present invention first provides a method, system and program product for replicating message status changes across (replica) messaging

systems. Specifically, under the present invention, when a status of a message for a user is changed on a first messaging system, a corresponding entry is made in a log associated with the user on the first messaging system. Thereafter, the entry is communicated to a second messaging system that also has a log associated with the user. Before the entry is formally replicated, however, it is first determined whether the status change is more recent than any other status changes that might exist for the same message with the log of the second messaging system. If so, the entry is replicated. If not, the entry is discarded from the second messaging system. In addition, the present invention provides a method, system and program product for maintaining log chronology for message status changes replicated across the systems. Specifically, when status change is entered into the log of the first messaging system, it is entered along with a clock time of the first messaging system. Thereafter, the entry containing the status change and the clock time is replicated to the second messaging system. If the entry is to be logged in the log of the second messaging system, it is logged along with the clock time of the first messaging system as well as a clock time of the second messaging system (i.e., that corresponds to the clock time of the first messaging system when the status change occurred).

[0022] It should be understood in advance that as used herein, the term “message” typically refers to an electronic message such as an electronic mail message, an instant message, etc. In any event, referring now to Fig. 1, an illustrative replication architecture 10 according to the present invention is shown. As depicted, architecture 10 depicts messaging systems 12A-C, which users 18 can access to perform message-related activities (i.e., read or unread messages). In a typical embodiment, messaging systems

12A-C are replicas of one another that function in an electronic messaging environment. For example, messaging systems 12A-C could be electronic mail servers that function within a network environment. In such as case, communication between messaging systems 12A-C could occur over any type of network such as the Internet, a local area network (LAN), a wide area network (WAN), a virtual private network (VPN), etc. As such communication could occur via a direct hardwired connection (e.g., serial port), or via an addressable connection that may utilize any combination of wireline and/or wireless transmission methods. Any conventional network connectivity, such as Token Ring, Ethernet, WiFi or other conventional communications standards could be utilized. Moreover, connectivity could be provided by conventional TCP/IP sockets-based protocol.

[0023] Regardless, as further shown, each messaging system 12A-C includes a storage unit/database 14A-C that contains a set (e.g., one or more) of logs 16A-C. Under the present invention, each user 18 typically has his/her own log 16A-C within each storage unit 14A-C. Logs 16A-C generally contain entries that reflect status changes of messages (i.e., message activity) for users 18. For example, if user “A” read message “X” via messaging system 12A, a corresponding entry will be made in the log 16A associated with user “A.” Similarly, if user “A” later marked the same message as “unread” via messaging system 12A, another entry would be made in the log 16A associated with user “A.”

[0024] In providing the architecture 10 shown in Fig. 1, the present invention also provides accurate replication of message status changes among messaging systems 12A-

C. Specifically, using the example above, when user “A” reads message “X” via messaging system 12A, and an entry is made in his/her associated log 16A, that status change is replicated to the other logs 16B-C associated with user “A” on messaging systems 12B-C. In providing such replication, the present invention takes steps to ensure that proper log “precedence” and “chronology” is maintained (as will be further described below).

[0025] Referring now to Fig. 2, a more detailed diagram of architecture 10 is shown. It should be understood that architecture 10 of Figs. 1 and 2 (e.g., the quantity of messaging systems 12A-C) is intended to be illustrative only and is not intended to limit the teachings herein. In any event, in Fig. 2, user 18 is shown communicating with messaging system 12A. It should be appreciated that although user 18 is shown directly accessing messaging system 12A, user 18 could alternatively communicate with messaging system 12A indirectly (i.e., over a network as described above) via a computerized “user system” (not shown) such as a personal computer, laptop hand held device, etc. In such a case, communication could occur in a client-server or server-server environment. Furthermore, although user 18 is shown as communicating specifically with messaging system 12A, it should be appreciated that user 18 could actually communicate with any of messaging systems 12A-C (as depicted in Fig. 1).

[0026] As further depicted, messaging system 12A generally comprises central processing unit (CPU) 22, memory 24, bus 26, input/output (I/O) interfaces 28, external devices/resources 30 and storage unit 14A. CPU 22 may comprise a single processing unit, or be distributed across one or more processing units in one or more locations, e.g.,

on a client and computer system. Memory 24 may comprise any known type of data storage and/or transmission media, including magnetic media, optical media, random access memory (RAM), read-only memory (ROM), a data cache, etc. Moreover, similar to CPU 22, memory 24 may reside at a single physical location, comprising one or more types of data storage, or be distributed across a plurality of physical systems in various forms.

[0027] I/O interfaces 28 may comprise any system for exchanging information to/from an external source. External devices/resources 30 may comprise any known type of external device, including speakers, a CRT, LCD screen, hand held device, keyboard, mouse, voice recognition system, speech output system, printer, monitor/display, facsimile, pager, etc. Bus 26 provides a communication link between each of the components in portal server and likewise may comprise any known type of transmission link, including electrical, optical, wireless, etc.

[0028] Storage unit 14A can be any system (e.g., database) capable of providing storage for information under the present invention. Such information could include, for example, logs 16A, etc. As such, storage unit 14A could include one or more storage devices, such as a magnetic disk drive or an optical disk drive. In another embodiment, storage unit 14A includes data distributed across, for example, a local area network (LAN), wide area network (WAN) or a storage area network (SAN) (not shown).

Although not shown, additional components, such as cache memory, communication systems, system software, etc., may be incorporated into messaging system 12A. In addition, it should also be appreciated that although not shown, messaging systems 12B-

C (and any “user system”) would likely include computerized components similar to messaging system 12A. Such components have not been shown for brevity purposes.

[0029] Shown in memory 24 of messaging system 12A is status system 34A. Status system 34A could be incorporated as part of a messaging program such as an electronic mail program that is loaded on messaging system 12A. In general, status system 34A-C includes log entry system 36A, replication system 38A, precedence system 40A, table maintenance system 42A and log purging systems 44A. Identical status systems 34B-C are also typically provided on messaging systems 12B-C (e.g., within memory). Status systems 34A-C are configured to provide accurate replication of message status changes across messaging systems 12A-C. For example, assume user 18 accesses message server 12A and reads message “X.” In such a case, log entry system 36A will enter the status change into the log 16A that is associated with user 18. In a typical embodiment, log entry system 36A will create an entry for the status change at the end of the log 16A. Thus, the last entries in log 16A will reflect the most message recent activity for user 18.

[0030] In entering the status change in the associated log 16A, log entry system will also ensure proper log chronology. Specifically, it could be the case that messaging systems 12A-C, although replicas, have clocks that are out of synch. For example, messaging system 12A could report a time of 1:00 PM, while the clocks of messaging systems 12B-C report times of 1:10 PM and 1:20 PM, respectively. For proper replication, it is important that the timing of status changes be accurately maintained so that recent activities will not be accidentally overridden/trumped by earlier activities. To ensure proper log chronology, all status changes are entered into logs 16A-C with a clock time of

a source messaging system sending the replication as well as a clock time of a destination system receiving the replication that corresponds to the time of the source messaging system when the status changed occurred. For messaging systems that are originating the replication operation (i.e., where there is no source messaging system) such as messaging system 12A in the example used herein, the status changed is entered along with only the clock time of originating messaging system. Thus, if user 18 read message “X” on messaging system 12A at 1:00 PM, an illustrative entry could appear as follows:

Message X (AT=1:00 PM, AAT=1:00PM)

[0031] Under the present invention, replication system 38A will periodically replicate logs 16A (or the entries therein) to other messaging systems 12B-C so that all logs 16A-C are identical (i.e., in synch) with each other. A similar replication occurs from other messaging systems 12B-C. Such replication can occur on a scheduled basis, or when a certain quantity of status changes get logged. Moreover, in replicating its logs 16A, messaging system 12A could replicate to all other messaging systems 12B-C, or just to a single messaging system (e.g., 12B) with an instruction for that messaging system (e.g., 12B) to then replicate the logs 16A to another messaging system (e.g., 12C). In the case of the latter, replication would occur in a tree or chain-like. In both scenarios, replication typically occurs on a “chunk” basis. Specifically, logs 16A (as well as 16B-C) are divided into chunks of entries. These chunks are then reviewed by replication system 38A so that only the entries/status changes that were entered into log 16A since the previous/last replication are replicated. This avoids excessive data transfer. In any event,

in communicating logs 16A to another messaging system, replication system 38A will communicate both the status changes as well as the times stored therewith.

[0032] Upon receipt of the status change for message “X,” log entry system 36B of status system 34B on messaging system 12B will first identify its own corresponding clock time and determine if it is different from the clock time of messaging system 12A. In the example, cited above, the clock time for messaging system 12B was ten minutes faster than messaging system 12A. Accordingly, the clock time for messaging system 12B will read 1:10 PM. This difference will be noted by the log entry system of system 12B.

However, before the status change is formally replicated into the corresponding log 16B for user 18, precedence system 38B of status system 34B will ensure the proper timing of the status change. For example, it could be the case that user 18 originally read message “X” on messaging system 12A, and then later marked the same message as unread on messaging system 12B before the replication occurred from messaging system 12A.

Thus, if the “read” activity were to be replicated into log 16B, it would be out of order and accidentally “trump” the “unread” activity for the message. Accordingly, before any log entry is created pursuant to a replication, precedence system 38B will determine if the status change is more recent than any other status changes for the message in the corresponding log 16B. If it is not, then the status change should be discarded from messaging system 12B (e.g., because it is “stale”).

[0033] If, however, the status change is more recent than any other status changes for the message stored in the associated log 16B, it will be entered into the appropriate log 16B by log entry system 36B. As indicated above, status changes are entered with the clock

times of the source messaging system (if any) and the destination messaging system so that log chronology is maintained. Thus, in entering the status change, log entry system 36B will enter the status change along with the clock times of messaging system 12A and messaging system 12B. Specifically, the destination messaging system will take the AAT of the source messaging system and adjust it based on the clock skew. Since the AAT for messaging system 12A was 1:00PM, and the time skew between messaging systems 12A and 12B was 10 minutes, log entry system 36B would add 10 minutes to the AAT of 1:00PM so that the status changes is entered as follows:

Message X (AT=1:00 PM, AAT=1:10 PM)

[0034] A similar process would occur when replication occurs to messaging system 12C. For example, assume that the status change is replicated to messaging system 12C. Just as with the replication to messaging system 12B, the status change would be communicated to messaging system 12A along with the clock time of messaging system 12A. Upon receipt, log entry system 26C would identify any time differences. As indicated above, when messaging system 12A is at 1:00 PM, messaging system 12C is at 1:20 PM. Thus, a twenty minute disparity is present. Thereafter, precedence system 40C would determine if the status change is more recent than any other status changes for message “X” in log 16C associated with user 18. If not, then the status change would be discarded from messaging system 12C. If, however, it was the most recent status change, then it would be entered into the log 16C associated with user 18 along with the clock time of the source messaging system (i.e., messaging system 12A) and the destination messaging system (i.e., messaging system 12C). As indicated above, this is accomplished

by taking the AAT of the source messaging system (i.e., messaging system 12A) and adjusting it based on the time skew. Since the AAT for messaging system 12A was 1:00PM, and the time skew between messaging systems 12A and 12C was 20 minutes, log entry system 36C would add 20 minutes to the AAT of 1:00PM so that the status changes is entered as follows:

Message X (AT=1:00 PM, AAT=1:20 PM)

[0035] As further shown in Fig. 2, each status system 34A-C also includes a table maintenance system 42A-C and a log purging system 44-C. Table maintenance systems 42A-C maintains unread tables 46A-C that identify all “unread” messages for the users. For example, when message “X” was originally received for user 18 on messaging system 12A, it was listed in unread table 46A. Assuming user 18 later reads the message, table maintenance system 42A will remove the listing of the message from the table. If the user 18 later marks the same message as “unread,” table maintenance system 42A will re-list the message in the table. In keeping an unread table, several options are possible. In a first embodiment, as shown, each messaging system 12A-C could have its own unread table 46A-C. In this embodiment, the table can be updated as replication occurs. For example, assume that user 18 later marked message “X” as “unread” on messaging system 12B. In this event, when the status change was replicated to messaging system 12A and entered into log 16A, table maintenance system 42A would make sure that the unread table 46A listed message “X.” In another embodiment, a single centrally located unread table could be maintained that all messaging systems 12A-C can interact with. Log purging systems 44A-C purge logs 16A-C of entries that are older than

predetermined time limit (e.g., 90 days). This prevents logs 16A-C from becoming too congested with out of date information.

[0036] It should be understood that the present invention can be realized in hardware, software, or a combination of hardware and software. Any kind of computer system(s) - or other apparatus adapted for carrying out the methods described herein - is suited. A typical combination of hardware and software could be a general purpose computer system with a computer program that, when loaded and executed, carries out the respective methods described herein. Alternatively, a specific use computer, containing specialized hardware for carrying out one or more of the functional tasks of the invention, could be utilized. The present invention can also be embedded in a computer program product, which comprises all the respective features enabling the implementation of the methods described herein, and which - when loaded in a computer system - is able to carry out these methods. Computer program, software program, program, or software, in the present context mean any expression, in any language, code or notation, of a set of instructions intended to cause a system having an information processing capability to perform a particular function either directly or after either or both of the following: (a) conversion to another language, code or notation; and/or (b) reproduction in a different material form.

[0037] Referring now to Fig. 3, a method flow diagram 100 of the status change replication process according to the present invention is shown. As depicted, first step S1 is to change a status of a message for a user on a first messaging system. Second step S2 is to enter the status change into a log associated with the user on the first messaging

system. Third step S3 is to communicate the status change to a second messaging system. Fourth step S4 is to determine, on the second messaging system, if the status change is more recent than any other status changes of the message within a log associated with the user on the second messaging system. If not, that status change is discarded from the second messaging system in step S5. If, however, the status change is more recent than other status changes for the message, it is entered into the log associated with the user on the second messaging system in step S6.

[0038] Referring now to Fig. 4 a method flow diagram 200 of the log chronology maintenance process according to the present invention is shown. As shown, first step D1 is to change a status of a message for a user on a first messaging system. Second step D2 is to enter the status change into a log associated with the user on the first messaging system with a clock time of the first messaging system. Third step D3 is to communicate the status change to a second messaging system. Fourth step D4 is to enter the status change into a log associated with the user on the second messaging system with the clock time of the first messaging system and a clock time of the second messaging system.

[0039] The foregoing description of the preferred embodiments of this invention has been presented for purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed, and obviously, many modifications and variations are possible. Such modifications and variations that may be apparent to a person skilled in the art are intended to be included within the scope of this invention as defined by the accompanying claims.